Wrapper Design for Testing IP Cores with Multiple Clock Domains

Qiang Xu and Nicola Nicolici
Department of Electrical and Computer Engineering
McMaster University, Ontario, L8S 4K1, Canada
Email: xuqiang@grads.ece.mcmaster.ca, nicola@ece.mcmaster.ca

Abstract

This paper addresses the testability problems raised by embedded cores with multiple clock domains. The proposed solution, based on a novel core wrapper architecture, shows how multi-frequency at-speed test response capture can be achieved using low-speed testers synchronized with high-speed on-chip generated clocks. Using experimental data, the trade-offs between the number of tester channels, testing time, area overhead and power dissipation are discussed.

1 Introduction

System-on-a-chip (SOCs) designs in telecommunications, networking and digital signal processing applications employ intellectual property (IP) cores operating at different clock rates. In addition, many embedded cores operate internally using multiple frequencies. For example, for the design reported in [11] all the cores have more than three clock domains. To illustrate a multiple-frequency core-based SOC, Figure 1 shows a simple hypothetical design that comprises three cores with three different *physical clocks*. In addition, Core 2 consists of modules (M1,M2,M3) operating at different frequencies (f1,f2,f3). A *physical clock* is a chip-level clock, e.g., it can come from an oscillator, a phase-locked loop (PLL) or recovered from a data stream. All the internal clocks generated from the same physical clock are considered to be a part of the same physical clock domain. The multi-frequency modules communicate one with each other through synchronization logic and/or first in first out (FIFO) memory blocks. Although multi-frequency embedded cores present advantages, such as reduced power and silicon area, because of the clock skew and synchronization problems they require special attention during test.

The objective of this paper is to provide a solution for at-speed multi-frequency core testing when test data is transferred using a low speed automatic test equipment (ATE). The key to the proposed solution is a novel core wrapper architecture used to synchronize the external tester channels with the core's internal scan chains in the shift mode, and provide at-speed test control in the capture mode. In the following section the relevant related work is overviewed.

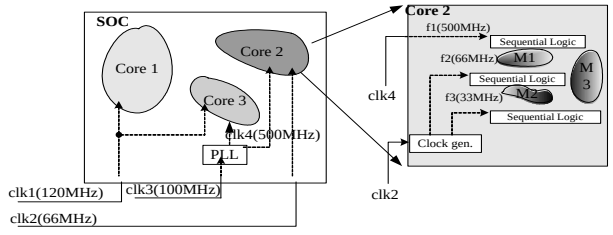

Figure 1. Multi-frequency SOC Example

1.1 Related Work

Embedded cores in an SOC are not directly accessible through the primary inputs, and consequently dedicated test access mechanisms (TAMs) are required to facilitate SOC testing. These TAMs are connected to the embedded cores using special interfaces called *core wrappers* [12]. Recently a number of approaches have addressed core wrapper design (e.g., [6–8, 10] only to name a few relevant ones). The work in [10] proposes a "test collar" as a test wrapper for SOC test. The method is based on two on-chip variable-width busses, one for test data and one for test control. Marinissen et. al [7], proposed a TestShell wrapper which is the basis for the IEEE P1500 [1] core wrapper. The TestShell is scalable and supports the operating modes required by the IEEE P1500. Wrapper design optimization to reduce testing time was proposed in [8] and [6] introduced an algorithm (based on a best fit decreasing heuristic) to minimize core testing time and required TAM width at the same time.

Since the existing core wrapper design approaches are applicable to single-frequency embedded core test, in the following an overview of the proposed approaches for multi-frequency testing is given. The main solutions are based on built-in self-test (BIST) [2, 3, 5, 9]. The solution presented in [3] used a dedicated clock generator to shift the multi-frequency scan chains at their corresponding clock frequencies. To avoid clock skew during capture retiming latches or dedicated two-phase/two-edge clocking schemes were used between different clock domains. In [9] the test data was shifted in/out at-speed by reusing the existing clock tree on chip and a programmable *scan mode* signal unit was used to control the capture of the circuit responses. The solutions proposed in [2, 5] employ a rather different approach that

1530-1591/04 $20.00 (c) 2004 IEEE

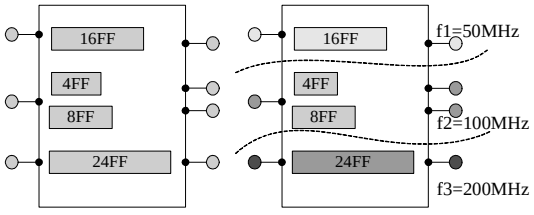

Figure 2. Single/Multiple Frequency Cores

separates the clocking for scan and capture in two phases, by multiplexing the clock signals for each phase. The main difference between these two approaches lies in the design of the *capture window*. In [2], in the capture mode, all the rated clocks are applied iteratively in a number of intervals equal to the number of clock domains. In each iteration the selected clock signal will propagate to a scan chain only if its value is lower than the rated clock of the respective scan chain (see [2] for detailed operation). In [5] a more flexible *capture window* was used, which consists of captures in different clock domains and some shift operations to create inter-domain at-speed capture. The functional clock of each of the domains is used to obtain a shift followed by a capture. In addition, shift operation for all the scan chains can work at any of the on-chip frequencies.

1.2 Motivation and Objectives

Despite the fact that BIST is the primary solution for at-speed multi-frequency testing [2, 3, 5, 9], there are a number of issues which arise in the SOC paradigm from the core provider and system integrator inter-operability perspective. There are four main cases: the system integrator will receive a (*i*) BIST-ed core, a (*ii*) BIST-ready core, a (*iii*) scan-testable core, or a (iv) functional-testable core. With the exception of the BIST-ed multi-frequency cores, in order to deliver the patterns (using *low-speed testers*) to the IP-protected cores and to perform rapid at-speed test (using *high-speed on-chip generated clocks*) without exceeding the power ratings or maximum shift frequency, the system integrator is constrained to design a multi-frequency core wrapper, which is the very aim of this paper. Unlike the existing approaches which assume that designs/cores can be BIST-ed for multi-frequency test, i.e., the structural netlist can be modified with extra hardware to guarantee valid multi-frequency capture, our approach is suitable for IP-protected cores where the SOC integrator is in charge of developing the multi-frequency test strategy.

The existing wrapper design algorithms developed for IP-protected cores (e.g., [6–8, 10]) are not directly applicable to at-speed multi-frequency testing because of the clock skew problem during test [5]. Although by grouping the flip-flops triggered by the same clock together and adding lock-up latches in between different clock domains, clock skew problem during shift can be solved (for mux-based scan approach), clock skew during at-speed capture still might occur and corrupt the test response. Therefore to solve the emerging problems for non-BISTed IP-protected multi-frequency embedded cores (i.e., determining a reliable and cost-efficient shift frequency and designing at-speed capture windows without any structural modifications to the core) we propose a novel core wrapper design algorithm and its associated new multi-frequency core wrapper architecture.

2 Multi-Frequency Core Wrapper (MFCW)

This section presents the multi-frequency core wrapper architecture and design algorithm, by focusing on the case when all the internal clocks are generated from a single physical clock. As summarized at the end of this section, the proposed solution can easily be extended to the general case when a core has multiple physical clocks. In order to describe the proposed architecture and algorithm the single-frequency core wrapper design (labeled as **SFCWD**) is extended to the multi-frequency core wrapper design problem (labeled as **MFCWD**), which is formulated as follows:

MFCWD: *Given a core with its test set parameters, i.e., the number of clock domains N_c, each clock domain group comprising i_g inputs, o_g outputs and s_g scan chains with given scan chain lengths, determine a wrapper design (including the architecture of the wrapper, the shift frequency and the capture window) with minimum testing time for the given TAM width constraint.*

To perform at-speed multi-frequency testing, although it is not necessary to load/unload data at functional frequencies, the last launch and capture must be done at-speed [5]. To facilitate this, the physical test clock needs to function at the highest frequency f_h during launch/capture. Since most of the available testers cannot provide test data at the highest on-chip frequency, we need to provide a mechanism which can use low-speed testers to transfer test data (shift phase), yet perform test application using high-speed on-chip functional multiple frequencies (capture phase). To achieve this the tester must synchronize with an on-chip low frequency f_l, derived from the on-chip high functional frequency f_h (e.g., coming from a PLL), which is used to shift in/out test data from/to the ATE. In the following the details of the proposed wrapper architecture and design algorithm are given.

Wrapper Scan Chains in Multiple Frequency Groups: A multiple frequency core is shown in Figure 2, where in addition to labeling the scan chains with their length, the associated normal operating frequency is also provided. Core wrapper design is mainly concerned with the construction of wrapper scan chains (*WSCs*) such that the testing time is minimized. The WSCs are composed from the input/output wrapper cells and the internal scan chains. When using single-frequency core wrappers, the testing time (in seconds) will be a function of the longest WSC [8] and the single shift frequency f, given be the following equation:

$$\tau(C) = \{(1+\max(wsc_i, wsc_o)) \times n_v + \min(wsc_i, wsc_o)\}/f \qquad (1)$$

where wsc_i/wsc_o are the lengths of the maximum input/output WSC respectively, and n_v is the number of test

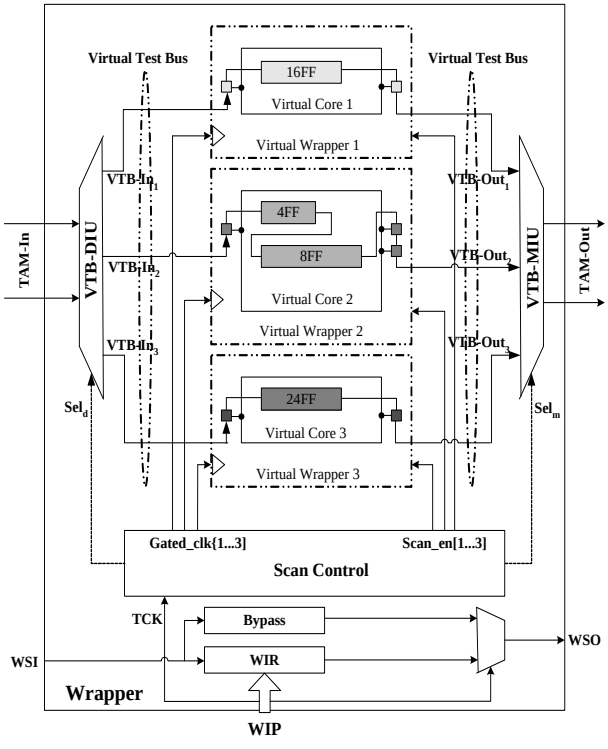

Figure 3. Multi-Frequency Core Wrapper

vectors in the test set for core C. For multi-frequency core wrappers care is taken not to combine scan chains belonging to different frequency groups, and consequently the testing time of the core is not only dependent on the lengths of the WSCs but also on the shift frequency f^g for different frequency groups (note, f^g is not necessarily the same as its functional frequency). For multi-frequency core test, since cores with different clock domain configuration will have different capture window design, hence different length of capture cycles, the testing time of the multi-frequency core can be formulated by the following equation:

$$\tau(C) = \max_g\{\max(wsc_i^g, wso_o^g) \times n_v + \min(wsc_i^g, wso_o^g)\}/f^g\} + t_c \times n_v \quad (2)$$

where the wsc_i^g / wsc_o^g are the lengths of the maximum input/output WSCs of group g (with shift frequency f^g) from core C, t_c is the time spent on capture phase for each pattern and n_v is the number of test vectors. To compute the WSCs for the multi-frequency core, a single-frequency algorithm can be employed and adapted for each frequency group, as shown later in this section.

Core Wrapper Interface: A multi-frequency core wrapper for the example core shown in Figure 2(b) is shown in Figure 3 (INTEST mode is illustrated). When compared to a P1500 single-frequency core wrapper (labeled as **SFCW**), the proposed multi-frequency core wrapper (labeled as **MFCW**) includes the same interface signals: serial input/output (WSI/WSO), parallel input/output (TAM-In/TAM-Out), and the wrapper interface port (WIP). The WIP provides test control and test clock for the core under test. Additional off-chip test clocks (i.e., ATE supplied)

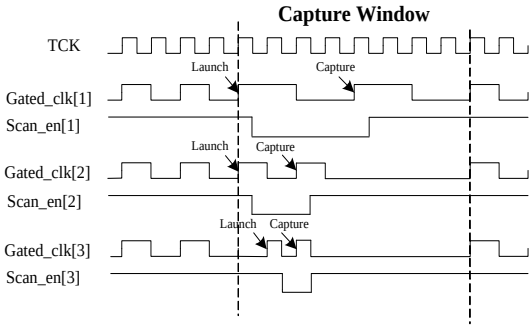

Figure 4. At-speed Multi-Frequency Testing Timing Diagram with a Single Physical Clock

or on-chip generated test clocks can also be included in the multi-frequency interface.

Core Wrapper Architecture: Although the MFCW interface is similar to the SFCW one, the architecture of the proposed MFCW differs significantly from the standard SFCW. Logic blocks belonging to different frequency domains are grouped and marked in the figure as *virtual cores*, and for each virtual core a *virtual wrapper* (single-frequency core wrapper), containing the WSCs for the respective group, is assigned. The virtual wrapper is connected to the interface through virtual test bus (VTB) lines. We assume that the system integrator uses parallel TAM-In/TAM-Out lines for MFCW's case (this is optional for SFCW). The TAM-In is connected to a *virtual test bus de-multiplexing interface unit (VTB-DIU)*, which drives data in the *virtual test bus lines*. Similarly, TAM-Out is connected to a *virtual test bus multiplexing interface unit (VTB-MIU)*, which collects the data from the virtual test buses (VTB-DIU and VTB-MIU are explained later on this section).

Scan Control Block: Scan Control block is a key part of the proposed MFCW, since it is used to generate the gated clocks (*Gated_clk*) necessary for shift and capture phases and scan enable signals (*Scan_en*) required by each virtual core in the capture phase. Since for at-speed testing, it is not necessary to load/unload test data at the rated frequencies, the shift frequency is used to tradeoff testing time against power dissipation. Unlike [4], we do not speed up the test data load/unload to its functional frequency through serializing/de-serializing technique since this will reduce the tester channel capacity and increase test power. Rather, we load/unload test data from/to the ATE at the speed of the tester frequency f_t and distribute it to multiple scan chains at the speed of shift frequency f_s using the proposed wrapper architecture. For each virtual core, we use the same frequency f_s during shift, which is switched to the functional frequency in the capture window (see Figure 4). However, the shift frequency f_s is not necessarily the same as the tester frequency f_t. For the example from Figure 2(b), if we assume the maximum tester frequency is 120MHz we will synchronize the tester with the on-chip frequency equal to $f_t = 100MHz$. The value of f_s and the number of the virtual test bus lines N_{vtb} depend

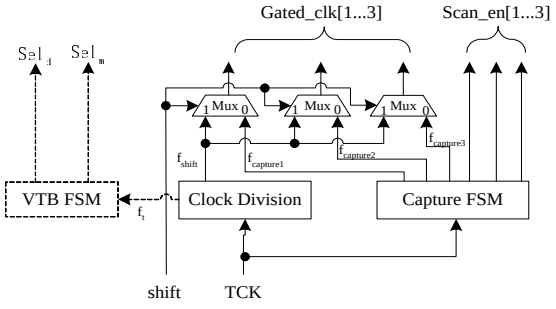

Figure 5. Scan Control Block

not only on tester frequency f_t and the available TAM width W_{tam}, but also on the number of the clock domains N_c, such that $N_{vtb} \geq N_c$ and $N_{vtb} \times f_s = f_t \times W_{tam}$ are satisfied. To decrease the power consumption during test, the objective is to find the lowest possible shift frequency f_s without affecting testing time. To simplify the hardware implementation we select the ratio of $\frac{f_t}{f_s}$ as two's exponent. For example, in Figure 3, the tester frequency f_t is selected to be 100MHz and if the available TAM width $W_{tam} = 2$, which is less than $N_c = 3$, then we will select f_s at 50MHz and the total number of virtual core test bus lines will be $N_{vtb} = \frac{100 \times 2}{50} = 4$. Although decreasing f_s to 25MHz, thus leading to $N_{vtb} = 8$, will reduce power consumption during test it will increase testing time, since having 8 VTB lines will be more than necessary for the 4 scan chains of the core from Figure 3.

By grouping flip-flops from the same clock domain into separate scan chains we eliminate the problem of clock skew during shift. However, to avoid the clock skew problem during capture we employ a capture window. When compared to [5], to adapt capture window to the core provider/system integrator model, the proposed solution is not programmable. However, its control is *embedded in the core wrapper architecture*. In addition, based on core provider's information, the transition-free clock domains [1] are captured at the same time (this decreases also test generation complexity), while the transition-hazard clock domains are captured at different times to avoid the inter-domain clock skew. This is achieved through carefully controlling the *Gated_clk* and *Scan_en* signals using the Capture FSM shown in Figure 5. The generation of these two signals justifies the use of the highest frequency physical clock as the core wrapper TCK signal. If the shift frequency f_s is lower than the tester frequency f_t then an internal control finite state machine (VTB FSM) is used to generate the mux select signal for VTB-DIU and VTB-MIU (see Figure 5). The testing timing diagram for the example from Figure 2(b) is shown in Figure 4. It should be noted that clock domains 2 and 3 are transition-free and hence they can be safely captured simultaneously, while the clock domain 1 will capture data at a different time to eliminate the test invalidation problem arising from clock skew during capture.

[1] If there are no data transfers between two clock domains or data transfers between two clock domains are safe during capture, then they are called *transition-free*. Otherwise, they are called *transition-hazard* clock domains.

Multi-Frequency Core Wrapper Design Algorithm - MFCWD

INPUT: C, W_{tam}, f_t, N_c
OUTPUT: f_s, $VC = \{VC^g | g = 1...h^j\}$, $SC = \{SC^v\}$

1. **Initialize** VC^g;
2. **Initialize** n to make $N_{vtb} \geq N_c$;
3. **while** (true) {
4. **Assign** $f_s = f_t \div n$, $N_{vtb} = W_{tam} \times n$;
5. **Assign** $N_{assigned\_vtb} = 0; VTB_{vc^g=0}$;
6. **for** i from 1 to N_{vc^g} {
7. **Assign** $VTB_{vc^i} = 1$;
8. $N_{assigned\_vtb}$++;
9. **do** SFCWD;
. }
10. **while** (true) {
11. **find** $g_{tat}^{\max} = \max\{\tau^g\}$;
12. VTB_g++; $N_{assigned\_vtb}$++;
13. **do** SFCWD;
14. **if**($N_{assigned\_vtb} == N_{vtb}$) **break**;
15. **if** no further reduction possible **break**;
. }
16. **if** (T_{shift} increase) **break**;
17. **Assign** $n = n \times 2$;
. }
18. **done**

VTB-DIU and VTB-MIU Blocks: VTB-DIU block is used to synchronize the input test data and to transfer the test vectors into the corresponding virtual cores. If the shift frequency f_s is selected to be the same as the tester frequency f_t, then we can simply connect the TAM lines to VTB lines through a flip-flop. However, if a lower shift frequency is used, then VTB FSM is needed to implement the de-multiplexing unit. It is obvious that by setting the Sel_d to a value, the data from the TAM lines is loaded into the corresponding VTB at f_s with a latency of one clock cycle. In addition, the last shift launch bit is registered in this block and it is shifted in at the correct time decided by the **Scan Control** block, which is an essential feature required by at-speed test through last shift launch. The multiplexing unit is the opposite of the demultiplexing unit, i.e., it is used to synchronize the output test data and transfer the test responses to the corresponding TAM lines. Note that, both VTB-DIU and VTB-MIU are active only in the test mode and hence do not infer any additional performance penalty when compared to the standard P1500 wrapper.

Multi-Frequency Core Wrapper Design (MFCWD): The wrapper design algorithm takes as inputs the tester frequency (f_t), the test parameters of core (C), the TAM width (W_{tam}) and the number of clock domains N_c, and it outputs the shift frequency (f_s) and the final wrapper design, including the virtual core wrapper and the wrapper scan chains. The algorithm initializes virtual cores (VC^g), by assigning to each

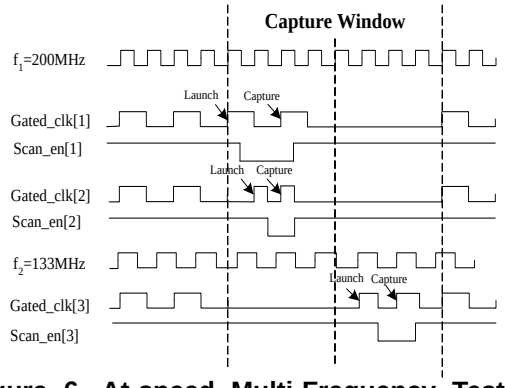

Figure 6. At-speed Multi-Frequency Testing Timing Diagram with Multi Physical Clocks

virtual core the inputs, scan chains and outputs which operate in its clock domain (line 1). In line 2 the ratio n ($\frac{f_t}{f_s}$) is initialized such that the number of virtual test bus lines N_{vtb} exceeds N_c. The algorithm loops through different configurations of f_s and N_{vtb} in order to reach the minimum shift time T_{shift} for a pattern. In line 5 the total number of assigned virtual test bus lines $N_{assigned\_vtb}$ and the number of assigned virtual test bus lines for each virtual core VTB_{vc^g} are initialized as zero. Each virtual core VC_g is first allocated one virtual test bus line and then the single frequency core wrapper design (SFCWD) is performed to get an initial testing time (lines 6-9) to be used as the starting point for virtual test bus line allocation (lines 10-15). Depending on N_{vtb}, the algorithm proceeds as follows. First, all the virtual cores are sorted based on their testing time (τ^g) and the virtual core which needs the longest testing time is identified (line 11). Then the following steps will iteratively assign virtual test bus lines to virtual cores. The basic idea is to assign more virtual test bus lines to the cores with longer testing time (line 12). Whenever a virtual test bus line is assigned, SFCWD is performed again to get the new testing time (line 13). There are two exit points for the inner loop: one when all the virtual test bus lines are assigned and all the virtual cores have been connected (line 14); the other exit point is when there is no further testing time reduction possible (line 15). Note, that the SFCWD algorithm used in our implementation is based on design_wrapper algorithm in [6]. When the shift time for the core with the new f_s has increased, the algorithm halts since further growth in N_{vtb} will only increase testing time (line 16).

(MFCW) With Multiple Physical Clocks: So far it was shown how a *new core wrapper architecture* can address at-speed multi-frequency core test with only one physical clock domain. For the general problem, where the core comprises several physical clock domains, we still divide the core under test into virtual cores belonging to different clock domains and we can still use the same shift frequency for all the virtual cores. The main difference, however, lies in the design of the capture window. To reach at-speed testing without corrupting test data, we propose to separate the capture window into several separate sub-capture windows, corresponding to each physical clock domain. The Scan Control block (with all the physical clocks connected to it) generates at-speed launch/capture signals for each virtual core. For example, consider an embedded core with 3 virtual cores VC_1, VC_2, VC_3, that operate at 100MHz, 200MHz and 133Mhz separately. Suppose $f_s = f_t = 100MHz$ (division of $f_{VC_2} = f_1$ = 200 MHz), then the timing diagram is shown in Figure 6, which shows two separate sub-capture windows: one for clock domains 1 and 2, which are transition-free, and one for clock domain 3.

$f(MHz)$	N_{in}	N_{out}	N_{bi}	N_{sc}	SC_{length}
200	38	42	0	5	100 100 100 98 98
100	24	29	32	3	88 88 87
133	34	10	0	1	76
50	42	62	0	4	96 96 64 62

Table 1. hCADT00 Clock Domain Information

3 Experimental Results

Since no existing approaches have tackled the multi-frequency embedded core testing problem, it is difficult to provide a one to one comparison to previous work. We have decided to analyze the trade-offs of the proposed solution, in terms of the number of tester channels, testing time, area overhead and power dissipation. Therefore, in this section, we present experimental results for a hypothetical multi-frequency core hCADT00. This core has four clock domains inside: the clock domain information is shown in Table 1, in which f denotes the functional frequency; N_{in}, N_{out}, N_{bi} and N_{sc} are the number of inputs, outputs, bidirectionals and scan chains in the specific clock domain respectively; and the length of each scan chain is shown in column SC_{length}. Note, this is a hard core, i.e., the internal scan chains cannot be divided, and the wrapper scan chains contain also the I/O boundary cells.

For different TAM widths W_{tam}, the shift frequency of the core f_s, virtual test bus lines assigned to each VC ($VTB[1...4]$), the necessary shift clock cycles C_{shift} and time T_{shift} for each pattern, and the additional area overhead Num_{gates} introduced by the new MFCW are shown in Table 2. In this experiment, we assume that the maximum frequency of the tester is 120MHz and, since we synchronize the tester with a division of the maximum internal frequency (200MHz), the tester will shift test data at $f_t = 100MHz$. From the experimental results shown in Table 2 we can observe that the shifting times for $W_{tam} = 24$ and $W_{tam} = 16$ are the same. This is because when the available TAM width exceeds 16 (which is the maximum number of wrapper scan chains), the shift time for virtual core VC_1 has already achieved its lowest value (100 clock cycles) and assigning more VTB lines to it will not lead to any improvements. It can also be seen that when the available TAM width is small (≤ 4 in Table 2), a lower shift frequency is selected. For example, when the available TAM width is 3, f_s is selected to be 25MHz and the total available VTB lines are $\frac{f_t \times W_{tam}}{f_s} = 12$.

It is interesting to note that by using lower shift frequency f_s not only the power consumption during test is reduced, but the testing time can also be decreased in several cases. In Table 3 power denotes the percentage of power consumption for the case when $f_t = f_s$ (only the dynamic power component is accounted for). When the available TAM width for hCADT00 is 4, if we select the shift frequency $f_s = f_t = 100MHz$, the shift time for each pattern will be 5.38 μs, which is larger than the shift time of 3.96 μs for $f_s = 50MHz$. This is because, the latter case allows for a better distribution of VTB lines to different clock domains. In addition, if a small increase in testing time is acceptable, the test power can be further reduced. For example, if we select the shift frequency $f_s = 25MHz$, the testing time will be 4 μs rather than 3.96 μs with $f_s = 50MHz$, however, the test power will be decreased by an additional factor of 2.

In terms of area overhead, VTB-DIU and VTB-MIU blocks need one flip-flop for each virtual test bus line and additional logic for multiplexing/demultiplexing if the shift frequency is lower than the tester frequency. The capture window size and the number of clock domains decide the hardware overhead of the scan control block. As it can be seen in the last column of Tables 2 and 3, the new MFCW will introduce an additional area overhead (for hCADT00) in the range of 190 to 344 equivalent 2 input NAND gates (2NANDs). This data is compiled using a 0.18 process technology where our results indicate that the overhead (for hCADT00) introduced by scan only is 1463 2NANDs and by scan and P1500 logic is 4512 2NANDs (this value is rather large since the number of I/Os is 313). Even if the number of I/Os will be lower, we believe that the added overhead to scan and P1500 logic will be in around 10%. For complex cores with hundreds of thousands of gates this is insignificant, when compared to the benefits of facilitating at-speed multi-frequency test of IP-protected cores.

4 Conclusion

This paper proposed a new core wrapper architecture, which, by means of a capture window, facilitates multi-frequency at-speed testing, while accepting data from a low-speed tester, at the expense of small on-chip area overhead. In addition, the power consumption during test is decreased by shifting data with lower frequency without penalizing the testing time. The proposed architecture provides a P1500 compatible solution for multi-frequency core test.

Acknowledgment: The authors wish to express their gratitude to Paul Theo Gonciari from Southampton University for motivating discussions and his insightful comments.

References

[1] IEEE P1500 Web Site. http://grouper.ieee.org/groups/1500/.

[2] S. Bhawmik. Method and apparatus for built-in self-test with multiple clock circuits. U.S. Patent 5680543, Lucent Technologies Inc., Murray Hill, N.J., Oct. 21, 1997.

W_{tam}	f_s (MHz)	$VTB[1...4]$	C_{shift} (cc)	T_{shift} (μs)	Num_{gates}
24	100	[6 4 2 4]	100	1	333
16	100	[6 4 2 4]	100	1	333
8	100	[3 2 1 2]	198	1.98	241
4	50	[3 2 1 2]	198	3.96	246
3	25	[5 3 1 3]	127	5.08	293
2	25	[3 2 1 2]	198	7.92	248
1	12.5	[3 2 1 2]	198	15.84	271

Table 2. hCADT00 Wrapper Design with Different TAM Width

f_s (MHz)	$VTB[1...4]$	C_{shift} (cc)	T_{shift} (μs)	$Power$ $(\%)$	Num_{gates}
100	[1 1 1 1]	538	5.38	100	190
50	[3 2 1 2]	198	3.96	50	246
25	[6 4 2 4]	100	4	25	338
12.5	[6 4 2 4]	100	8	12.5	344

Table 3. hCADT00 Wrapper Design ($W_{tam} = 4$)

[3] B. N. Dostie, A. S. Hassan, D. M. Burek, and S. K. Sunter. Mulitple clock rate test apparatus for testing digital systems. U.S. Patent 5349587, Northern Telecom Limited, Montreal, Canada, Sep. 20, 1994.

[4] D. Heidel, S. Dhong, P. Hofstee, M. Immediato, K. Nowka, J. Silberman, and K. Stawiasz. High speed serializing/deserializing design-for-test method for evaluating a 1 ghz microprocessor. In *Proc. 16th IEEE VLSI Test Symposium*, pages 234 –238, 1998.

[5] G. Hetherington, T. Fryars, N. Tamarapalli, M. Kassab, A. Hassan, and J. Rajski. Logic BIST for large industrial designs: Real issues and case studies. In *Proc. International Test Conference*, pages 358–367, 1999.

[6] V. Iyengar and K. Chakrabarty. Integrated wrapper/TAM co-optimization, constraint-driven test scheduling, and tester data volume reduction for SOCs. In *Proc. IEEE/ACM Design Automation Conference*, pages 685–690, 2002.

[7] E. J. Marinissen et al. A Structured And Scalable Mechanism for Test Access to Embedded Reusable Cores. In *Proceedings IEEE International Test Conference (ITC)*, pages 284–293, Washington, DC, Oct. 1998.

[8] E. J. Marinissen, S. K. Goel, and M. Lousberg. Wrapper Design for Embedded Core Test. In *Proceedings IEEE International Test Conference (ITC)*, pages 911–920, Atlantic City, NJ, Oct. 2000.

[9] B. Nadeau-Dostie, D. Burek, and A. S. M. Hassan. ScanBist: A multifrequency scan-based BIST method. *IEEE Design and Test of Computers*, pages 7–17, Spring 1994.

[10] P. Varma and S. Bhatia. A Structured Test Re-Use Methodology for Core-Based System Chips. In *Proceedings IEEE International Test Conference (ITC)*, pages 294–302, Washington, DC, Oct. 1998.

[11] B. Vermeulen, S. Oostdijk, and F. Bouwman. Test and Debug Strategy of the PNX8525 Nexperia^{TM} Digital Video Platform System Chip. In *Proceedings IEEE International Test Conference (ITC)*, pages 121–130, Oct. 2001.

[12] Y. Zorian, S. Dey, and M. Rodgers. Test of future system-on-chips. In *Proceeding of the IEEE/ACM International Conference on Computer-Aided Design*, pages 392–398, 2000.